@[toc]

摘要

Click-through rate prediction is an essential task in industrial applications, such as online advertising. Recently deep learning based models have been proposed, which follow a similar Embedding & MLP paradigm. In these methods large scale sparse input features are first mapped into low dimensional embedding vectors, and then transformed into fixed-length vectors in a group-wise manner, finally concatenated together to fed into a multilayer perceptron (MLP) to learn the nonlinear relations among features. In this way, user features are compressed into a fixed-length representation vector, in regardless of what candidate ads are. The use of fixed-length vector will be a bottleneck, which brings difficulty for Embedding & MLP methods to capture user's diverse interests effectively from rich historical behaviors. In this paper, we propose a novel model: Deep Interest Network (DIN) which tackles this challenge by designing a local activation unit to adaptively learn the representation of user interests from historical behaviors with respect to a certain ad. This representation vector varies over different ads, improving the expressive ability of model greatly. Besides, we develop two techniques: mini-batch aware regularization and data adaptive activation function which can help training industrial deep networks with hundreds of millions of parameters. Experiments on two public datasets as well as an Alibaba real production dataset with over 2 billion samples demonstrate the effectiveness of proposed approaches, which achieve superior performance compared with state-of-the-art methods. DIN now has been successfully deployed in the online display advertising system in Alibaba, serving the main traffic.

1. 背景

Deep Interest Network(DIN)是盖坤大神领导的阿里妈妈的精准定向检索及基础算法团队，在2017年6月提出的。 它针对**电子商务领域(e-commerce industry)**的CTR预估，重点在于**充分利用/挖掘用户历史行为数据中的信息**。

数据特征： 针对互联网电子商务领域，数据特点：Diversity、Local Activation。

针对问题：

用户有多个兴趣爱好，访问了多个good_id，shop_id。为了降低纬度并使得商品店铺间的算术运算有意义，我们先对其进行Embedding嵌入。那么我们如何对用户多种多样的兴趣建模那？使用**Pooling对Embedding Vector求和或者求平均**。同时这也解决了不同用户输入长度不同的问题，得到了一个固定长度的向量。这个向量就是用户表示，是用户兴趣的代表。

但是，直接求sum或average损失了很多信息。所以稍加改进，针对不同的behavior id赋予不同的权重，这个权重是由当前behavior id和候选广告共同决定的。这就是Attention机制，实现了Local Activation。

DIN给出了解决方案：

1. 使用 **用户兴趣分布(Diversity)** 来表示用户多种多样的兴趣爱好
2. 使用 **Attention机制** 来实现Local Activation
3. 针对模型训练，提出了 **Dice激活函数，自适应正则(Adaptive Regulation)**，显著提升了模型性能与收敛速度

1.1 名词解释

这两个词在论文中通篇出现，先把其表示的意思说清楚。

Diversity： 用户在访问电商网站时会对多种商品都感兴趣。也就是用户的兴趣非常的广泛。

Local Activation： 由于用户兴趣的多样性，只有部分历史数据会影响到当次推荐的物品是否被点击，而不是所有的历史记录。

举个简单的例子，观察下面的表格：

Table 1: Examples of user behavior history from online product.

User	Behavior History	Candidate Ad
Young Mother	woolen coat, T-shirts, earrings,children's coat leather handbag, miniskirt, sports underwear	long sleeved jacket
Swimmer	bathing suit, kickboard, swimming cap, travel book tent, potato chips, nuts, potato chips, ice cream	goggle

Diversity体现在年轻的母亲的历史记录中体现的兴趣十分广泛，涵盖羊毛衫、手提袋、耳环、童装、运动装等等。而爱好游泳的人同样兴趣广泛，历史记录涉及浴装、旅游手册、踏水板、马铃薯、冰激凌、坚果等等。

Local activation体现在，当我们给爱好游泳的人推荐goggle(护目镜)时，跟他之前是否购买过薯片、书籍、冰激凌的关系就不大了，而跟他游泳相关的历史记录如游泳帽的关系就比较密切。

针对上面提到的用户行为中存在的两种特性，阿里将其运用于自身的推荐系统中，推出了深度兴趣网路DIN，接下来，我们就一起来看一下模型的一些实现细节。

1.2 相关工作

CTR预估是一个比较窄的研究领域，但是模型性能一点点的提升，在实际应用中都非常关键，真金白银毫不含糊。随着深度学习在CV、NLP等领域取得突破性进展，一些研究也开始尝试将DNN应用于CTR预估，比如：Wide&Deep, DeepFM等。

> 这些做法一般分为两部分：
>
> 1. 在输入上面加一层embeding层，把最原始高维度、稀疏的数据转换为低维度的实值表示上(dense vector)。
> 2. 增加多个全连接层，学习特征之间的非线性关系。 Sparse Features -> Embedding Vector -> MLPs -> Output

这些方法的优点在于：**相比于原来的Logistic Regression方法，大大减少了人工特征工程的工作量。**

缺点：在电子商务领域中，用户的历史行为数据(User Behavior Data)中包含大量的用户兴趣信息（**diversity**），而只有一小部分的用户兴趣信息会影响行为（click or not to click），即文中的**Local Activation**。

之前的研究并没有针对Behavior data features(multi-hot) **特殊的结构(Diversity + Local Activation)** 进行建模。

针对**Diversity**： 针对用户广泛的兴趣，DIN用 *an interest distribution*去表示。

针对**Local Activation**： DIN借鉴机器翻译中的Attention机制，设计了一种*attention-like network structure*， 针对当前候选Ad，去局部的激活(*Local Activate*)相关的历史兴趣信息。和当前候选Ad相关性越高的历史行为，会获得更高的*attention score*，从而会主导这一次预测。

针对大规模稀疏数据的模型训练：

当DNN深度比较深(参数非常多)，输入又非常稀疏的时候，很容易过拟合。DIN提出**Adaptive regularizaion**来防止过拟合，效果显著。

此外，引入dice activation function进一步加强了模型的表达效果。

论文还提出，DIN方法也可以应用于其他**有丰富用户行为数据**的场景，比如：

- 电子商务中的个性化推荐
- 社交网络中的信息推流排序(feeds ranking)

2. 系统总览

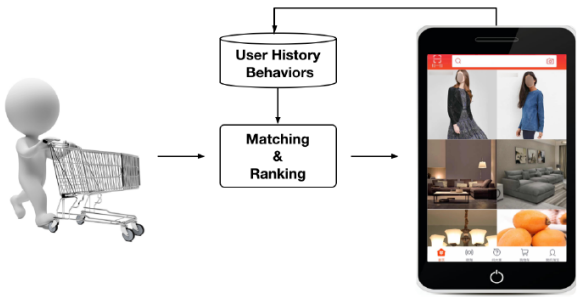


Figure 1: Illustration of running procedure of display advertising system in Alibaba, in which user behavior data plays important roles.

阿里推荐系统工作流程就像上图所示：

1. 检查用户历史行为数据
2. 使用matching module产生候选ads
3. 通过ranking module得到候选ads的点击概率，并根据概率排序得到推荐列表
4. 记录下用户在当前展示广告下的反应(点击与否)

这是一个闭环的系统，对于用户行为数据(User Behavior Data)，系统自己生产并消费。

2.1 电商CTR数据特点

前面提到，电子商务领域，充分利用User Behavior Data非常关键，而它又有着非常显著的特点：

- Diversity. 兴趣爱好非常广泛
- Local Activation. 历史行为中部分数据主导是否会点击候选广告

还有的特点，就是CTR中输入普遍存在的特点：

- 高纬度
- 非常稀疏

CTR中一旦涉及到用户行为数据，还有一个特点：

- 特征往往都是**multi-hot**的稀疏ids。

也就是：多值离散特征。比如：用户在YouTube上看的视频和搜索过的视频。无论是看过的还是搜索过的，都不止一个，但是相对于所有的视频来说，看过和搜索过的数量都太小了(非常稀疏)。 在电子商务上的例子就是：用户购买过的good_id有多个，购买过的shop_id也有多个，而这也直接导致了每个用户的历史行为id长度是不同的。

为了得到一个固定长度的Embedding Vector表示，原来的做法是在`Embedding Layer`后面增加一个`Pooling Layer`。Pooling可以用sum或average。最终得到一个固定长度的`Embedding Vector`，是用户兴趣的一个抽象表示，常被称作`User Representation`。缺点是会损失一些信息。

DIN使用Attention机制来解决这个问题。**Attention机制**来源于`Neural Machine Translation(NMT)`。DIN使用Attention机制去更好的建模局部激活。在DIN场景中，针对不同的候选广告需要自适应地调整`User Representation`。也就是说：在`Embedding Layer -> Pooling Layer`得到用户兴趣表示的时候，依据给定Ad，通过计算用户历史行为与该给定Ad的相关性，赋予不同的历史行为不同的权重，实现局部激活。从最终反向训练的角度来看，就是根据当前的候选广告，来反向的激活用户历史的兴趣爱好，赋予不同历史行为不同的权重。

2.2 特征处理(User Behavior Features)

参考论文`Learning piece-wise linear models from large scale data for ad click prediction`中*common feature trick*，目的是降低空间开销和计算开销。大体的思想是：同一个用户多条样本，它们之间很多信息重复，比如用户统计信息，昨天之前的统计信息等。针对这些重复信息只存储一次，并建立索引。

另外，论文中作者把特征分为四大类，并**没有进行特征组合/交叉特征**。而是通过DNN去学习特征间的交互信息。特征如下：

Table 2: Feature Representations and Statistics in our display advertising system.

Feature Category	Feature Name	Dimemsion	Type	#Nonzero Ids/Sample
User Profile Features	gender	2	one-hot	1
	age_level	~ 10	one-hot	1
	...	...	...	...
User Behavior Features	visited good_ids	$\sim 10^9$	multi-hot	$\sim 10^3$
	visited shop_ids	$\sim 10^7$	multi-hot	$\sim 10^3$
	visited cate_ids	$\sim 10^4$	multi-hot	$\sim 10^2$
	...	...	...	...
Ad Features	good_id	$\sim 10^7$	one-hot	1
	shop_id	$\sim 10^5$	one-hot	1
	cate_id	$\sim 10^4$	one-hot	1
	...	...	...	...
Scene Features	pid	~ 10	one-hot	1
	time	~ 10	one-hot	1
	...	...	...	...

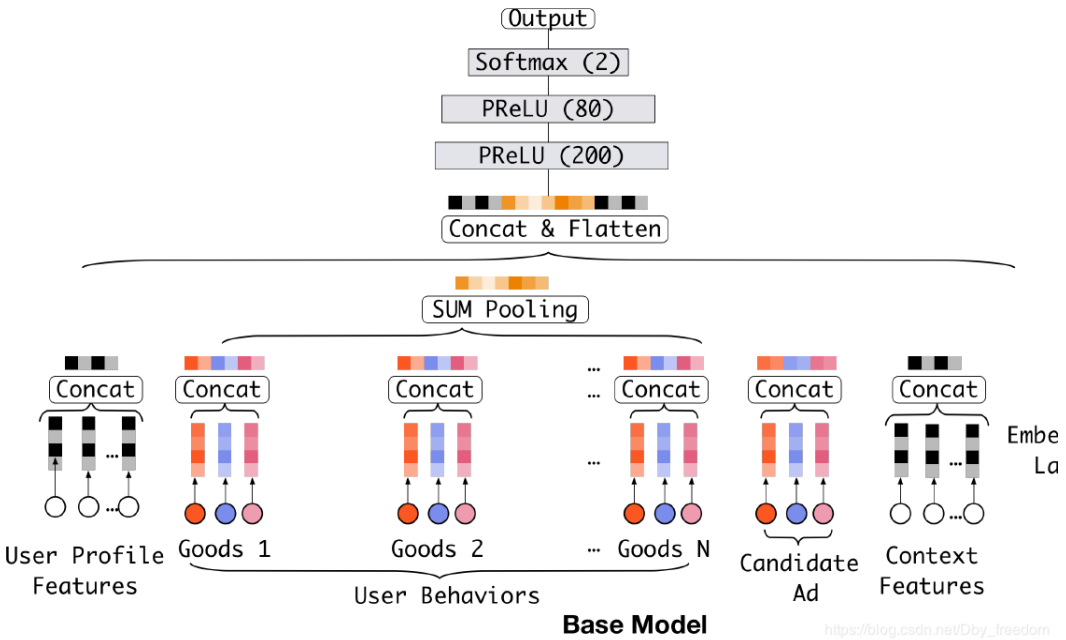
可以看到特征主要包括：用户特征、用户行为特征、广告特征、上下文特征。 其中，只有**用户行为特征**中会出现`multi-hot`，原因就是一个用户会购买多个good_id,也会访问多个shop_id，另一个表现就是这样导致了每个用户的样本长度都是不同的。针对这种特征，由于每个涉及到的非0值个数是不一样的，常见的做法就是将id转换成embedding之后，加一层pooling层，比如average-pooling，sum-pooling，max-pooling。

DIN中使用的是weighted-sum，其实就是加权的sum-pooling，权重经过一个activation unit计算得到。（设计一个attention-like network structure来关注那些和当前 ad 相关的历史行为信息。和候选 ad 相关性高的历史行为得到更多的attention score，从而主导这次预测。）也因此，不同ad下的**用户行为特征**表示不同（只对User Behaviors特征进行了attention机制）。

> 原因是用户行为特征的 **diversity** 特性，导致使用embedding + pooling方法得到的固定长度的embedding限制了用户行为特征的挖掘（这种情况下，无论来什么广告，用户行为特征都是同一个固定长度的embedding，而Ad是否点击却只有用户行为特征中的一小部分有关系），这种情况下得到的模型粒度较粗，模型表现性能因为User Behavior的定长embedding受限；这也正是为什么该论文提出attention机制的原因，通过引入attention机制，用户行为特征的表示不再是一个不变的定长embedding，而是通过与Ad特征进行关联，利用attention机制挖掘出与该ad最相关的用户历史行为进行加权表示，每个ad对应的user behavior特征不同，这样更进一步地细粒度地利用了user behavior特征（结合user behavior特征对电商广告系统是最重要的特征属性），极大地提升了模型表现。

2.3 BaseModel

在介绍DIN之前，我们先来看一下一个基准模型，结构如下：



如上图所示，Base Model主要由两部分组成：

1. 把稀疏的输入(id特征)转换成embedding vector
2. 增加MLPs得到最终的输出

Embedding layer:

输入是高维二进制矢量，通过embedding layer转换成低纬dense representations.

ont hot & multi-hot 示例：

$$\underbrace{[0,0,0,0,1,0,0]}_{\text{weekday=Friday}}\quad\underbrace{[0,1]}_{\text{gender=Female}}\quad\underbrace{[0,..,1,...,1,...0]}_{\text{visited\_cate\_ids=\{Bag,Book\}}}\quad\underbrace{[0,..,1,...,0]}_{\text{ad\_cate\_id=Book}}$$

one-hot、multi-hot示例，上图中，weekday, gender, ad_cate_id都是one-hot 编码，visited_cate_ids是multi-hot 编码。

对于第i个特征组 $t_i, W^i = [w_1^i, \dots, w_j^i, \dots w_{K_i}^i] \in \mathbb{R}^{D \times K_i}$ 表示第 i 个embedding dictionary ，其中 $w_j^i \in R^D$ 是一个维度为D 的embedding vector。

- 如果 t_i 是一个one-hot vector，其中第j个元素$t_i[j] = 1$，则embeded representation t_i 是一个single embeddding vector $e_i = w_j^i$
- 如果 t_i 是一个multi-hot vector，有 $t_i[j] = 1$ for $j \in \{i_1, i_2, \dots, i_k\}$，则 t_i 的 embeded representation 是一组embedding vectors: $\{e_{i_1}, e_{i_2}, \dots, e_{i_k}\} = \{w_{i_1}^i, w_{i_2}^i, \dots, w_{i_k}^i\}$

从embedding layer不难发现，对于user behavior feature，其经常出现为multi-hot 形式，经过embedding layer之后对应多个multi-hot vector，这也是为什么引入pooling layer的原因，这里base model 通过sum pooling将embeded multi-hot vector转换为ont -hot vector。

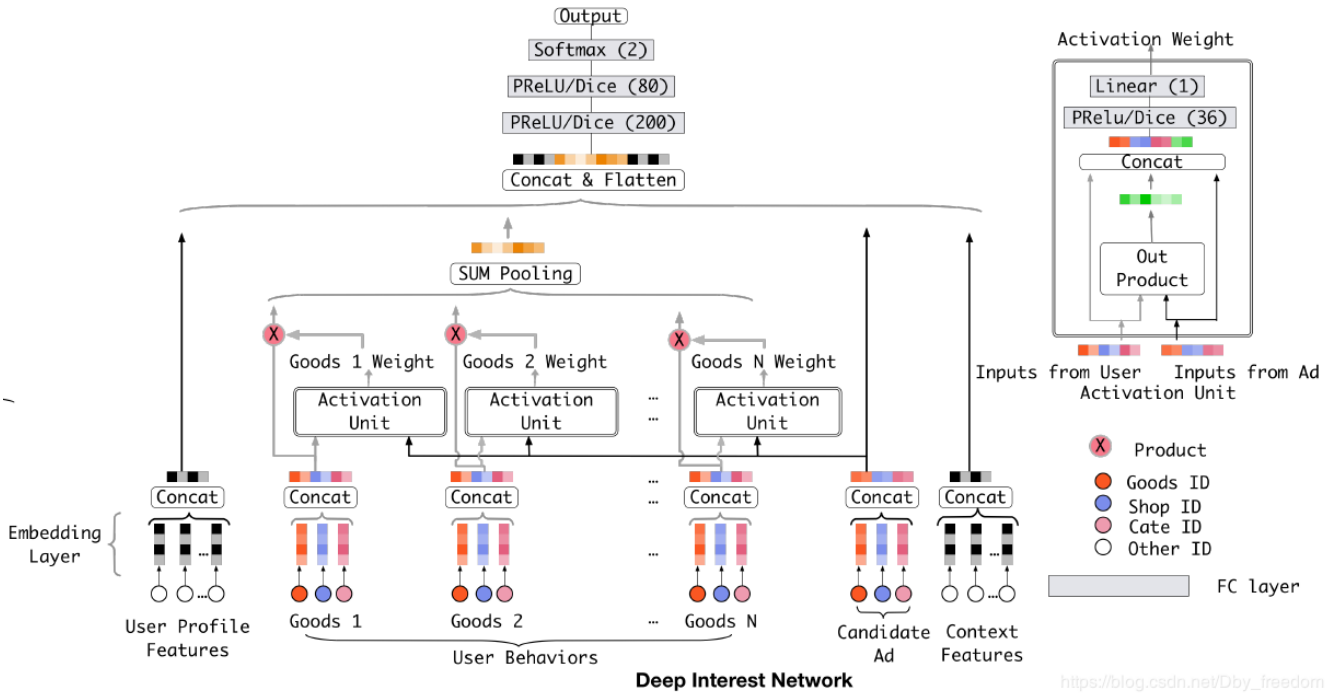
即对于一个用户，之前购买过的good_ids组成了一个`user behavior sequence ids`。针对不同的用户，这个序列的长度是不同的(不同用户购买的物品数量不同). 所以在Embedding Layer和MLPs中间，增加了一个**Pooling Layer**，使用的是**sum** operation，把这些goods或shops的embedding vector相加，这样，不管特征中有多少个非0值，经过转换之后的长度都是一样的，得到一个固定长度的向量作为MLPs的输入。

Base Model上线后表现很好，现在也在承担着阿里线上广告展示系统的大部分流量。（论文发表时）

> 好吧，博客中关于user behavior features 的multi-hot 进行embedding，再进行为sum-pooling得到固定长度的embedding操作讲述了很多遍。。实在是该DIN其本质是解决这个问题的（local activation unit)， 而另外两点创新Dice 激活函数 与 自适应正则(adaptive regulation) 都只是辅助模型训练的手段。

2.4 Deep Interest Network

Base Model有一个很大的问题，它对用户的历史行为是同等对待的，没有做任何处理，这显然是不合理的。一个很显然的例子，离现在越近的行为，越能反映你当前的兴趣。因此，对用户历史行为基于Attention机制进行一个加权，阿里提出了深度兴趣网络（Deep Interest Network)，先来看一下模型结构：



先给出结论：

1. Activation Unit实现Attention机制，对Local Activation建模（与传统的attention机制差别是舍弃了顶层的softmax，实现依据ad对用户行为特征进行权重判定）
2. Pooling(weighted sum)对Diversity建模（实现不同广告对应不同用户行为特征embedding weighted sum）

Attention机制简单的理解就是，针对不同的广告，用户历史行为与该广告的权重是不同的。假设用户有ABC三个历史行为，对于广告D，那么ABC的权重可能是0.8、0.4、0.1；对于广告E，那么ABC的权重可能是0.3、0.9、0.1（这里的权重和不一定为0，因为此论文中使用的activate unit中将传统attention机制里的softmax层去掉，如上图2右上角所示，因此不再保证有$\sum_i w_i = 1$）。这里的权重，就是Attention机制即上图中的Activation Unit所需要学习的。

之所以引入Local activation实现 attention 机制的原因是，如果不用Local activation的话，将会出现下面的情况：假设用户的兴趣的Embedding是 V_u，候选广告的 Embedding 是 V_a，用户兴趣和候选的广告的相关性可以写作 $F(U, A) = V_a * V_u$。如果没有Local activation机制的话，那么同一个用户对于不同的广告，V_u 都是相同的。举例来说，如果有两个广告A和B，用户兴趣和A，B的相似性都很高，那么在 V_a 和 V_b 连线上的广告都会有很高的相似性。这样的限制使得模型非常难学习到有效的用户和广告的embedding表示。

在加入Activation Unit之后，用户的兴趣表示计算如下：

$$V_u(A) = f(v_A, e_1, e_2, \ldots, e_H) = \sum_{j=1}^{H} \alpha(e_j, v_A) e_j = \sum_{j=1}^{H} w_j e_j$$

其中，$f(v_A, e_1, e_2, \ldots, e_H)$ 表示长度为H的用户U的用户行为embedding vector， V_A 是ad A的embedding vector，这样，$V_u(A)$ 的长度随着广告的不同而变化（长度不变，依旧为H），$a(\cdot)$ 是前馈神经输出为激活权重，如上图所示。

Activation unit 的输入一共有三项：

1. user behavior embedding vector;
2. ad embedding vector;

3. user behavior embedding vector 与 ad embedding vector的out product(外积)

加入两者的out product，是外积作为一种explicit knowledge，有助于相关性建模。

值得注意的是，这里 activation unit实现的attention机制相较于传统的attention机制，抛弃了顶层的softmax，也就是说不再保证 $\sum_{j=1}^{H} = 1$，而是将其作为用户行为特征对对应ad的相关程度，ad对应下的用户行为特征embedding加权和越大，其相关程度越强。

3. 训练技术

3.1 Mini-batch Aware Regularization

CTR中输入稀疏而且维度高，论文中举出features of goods_ids 维度为0.6 billion，若是不添加任何正则的话，模型表现在一个epoch之后快速下降，如下图所示：

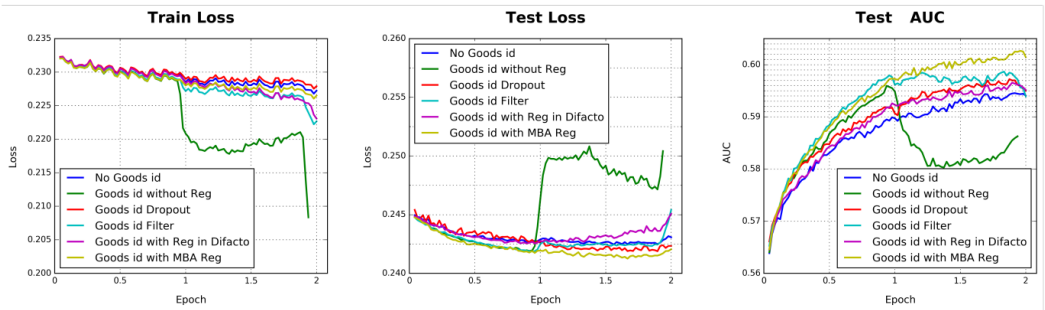


Figure 4: Performances of BaseModel with different regularizations on Alibaba Dataset. Training with fine-grained *goods_ids* features without regularization encounters serious overfitting after the first epoch. All the regularizations show improvement, among which our proposed mini-batch aware regularization performs best. Besides, well trained model with *goods_ids* features gets higher AUC than without them. It comes from the richer information that fine-grained features contained.

通常的做法是加入L1、L2防止过拟合，但这种正则方式对于工业级CTR数据不适用，结合其稀疏性及上亿级的参数，以L2正则化为例，需要计算每个mini-batch下所有参数的L2-norm，参数上升至亿级之后计算量太大，不可接受。

用户数据符合长尾定律long-tail law，也就是说很多的feature id只出现了几次，而一小部分feature id出现很多次。这在训练过程中增加了很多噪声，并且加重了过拟合。

对于这个问题一个简单的处理办法就是：直接去掉出现次数比较少的feature id。但是这样就人为的丢掉了一些信息，导致模型更加容易过拟合，同时阈值的设定作为一个新的超参数，也是需要大量的实验来选择的。

因此，阿里提出了**自适应正则**的做法，即： 1.针对feature id出现的频率，来自适应的调整他们正则化的强度； 2.对于出现频率高的，给与较小的正则化强度； 3.对于出现频率低的，给予较大的正则化强度。

提出mini-batch aware regularizer，只计算出现在mini-batch中的稀疏特征参数的L2-norm（即只计算mini-batch中非零项的L2-norm）。

给出简单推导，具体推导参考原论文：

$$L_2(W) \approx \sum_{j=1}^{K} \sum_{m=1}^{B} \frac{\alpha_{mj}}{n_j} ||w_j||_2^2$$

进一步推导得到近似的mini-batch aware version of L2 regularization，对第m轮mini-batch来说，特征 j 的嵌入权重是：

$$w_j \leftarrow w_j - \eta [\frac{1}{|B_m|} \sum_{(x,y) \in B_m} \frac{\partial L(p(x), y)}{\partial w_j} + \lambda \frac{\alpha_{mj}}{n_j} w_j]$$

其中 B_m 表示第 m 轮mini-batch，其中 $a_{ji} = max_{(x,y) \in B_m} I(x_j \neq 0)$ 表示在mini-batch B_m 中至少一个实例具有特征id j，w_j 表示j-th embedding vector，n_j 表示feature id j 在所有样本中的出现次数。

3.2 Dice: Data Dependent Activation Function

PReLU其实是ReLU的改良版，ReLU可以看作是 `x * Max(x,0)`，相当于输出x经过了一个在0点的阶跃整流器。由于ReLU在x小于0的时候，梯度为0，可能导致网络停止更新，PReLU对整流器的左半部分形式进行了修改，使得x小于0时输出不为0。 研究表明，PReLU能提高准确率但是也稍微增加了过拟合的风险。PReLU形式如下：

$$f(s) = \begin{cases} s, & if\ s > 0 \\ \alpha s, & if\ s \leq 0 \end{cases} = p(s) \cdot s + (1 - p(s)) \cdot \alpha s$$

其中 s 是激活函数的以为输入，其中$p(s) = I(s > 0)$ 是一个指示函数，控制 $f(s)$ 在 $f(s) = s$ 与 $f(s) = \alpha s$ 之间进行转换， α 是一个学习参数。

这里不妨把 $p(s)$ 当做控制函数，则该控制函数的图像为：

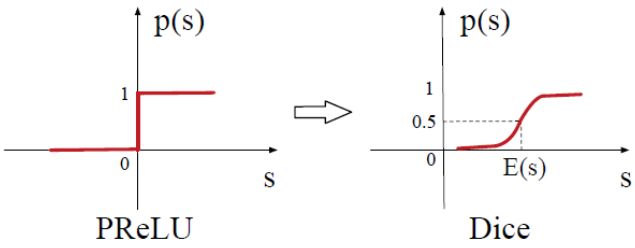


Figure 3: Control function of PReLU and Dice.

即PReLU函数以0点作为控制转折点，这对于输入层具有不同分布的情况不适用，因此，该论文设计了一种新型data adaptive activation function named **Dice**.

$$f(s) = p(s) \cdot s + (1 - p(s)) \cdot \alpha s,\ p(s) = \frac{1}{1 + e^{-\frac{s - E[s]}{Var[s] + \epsilon}}}$$

其控制函数$p(s)$ 如上图， 在训练阶段，$E[s]$ 和 $Var[s]$ 是每个mini-batch 的平均值和方差，在测试阶段，$E[s]$ 和 $Var[s]$ 随着数据进行移动， ϵ是一个小常数项，设定为10^{-8}。

其中Dice可以看做是PReLU的一种推广，主要思想是依据输入数据分布进行自适应调整修正点，该修正点不再默认为0，而是设定为数据均值；其次，Dice的一个好处是平滑过渡两个状态。

当$E(s) = 0$ and $Var[s] = 0$ 时，Dice退化为 PReLU。

4. 实验结果

4.1 训练数据

Table 2: Statistics of datasets used in this paper.

Dataset	Users	Goods[a]	Categories	Samples
Amazon(Electro).	192,403	63,001	801	1,689,188
MovieLens.	138,493	27,278	21	20,000,263
Alibaba.	60 million	0.6 billion	100,000	2.14 billion

[a] For MovieLens dataset, goods refer to be movies.

4.2 Competitors

- LR
- BaseModel: 如2.2所述
- Wide&Deep: 依据文献10进行cross-product of user behaviors and candidates as wide inputs.
- PNN: PNN 可以视为BaseModel的升级版，引入了product layer after embedding layers to capture high-order feature interactions.
- DeepFM: 即利用FM代替wide & deep 模型的wide部分

4.3 Metircs

使用了一种"An variation of user weighted AUC" 用于度量用户间order，通过平均每个用户的AUC，公式如下：

$$AUC = \frac{\sum_{i=1}^{n} \#impression_i \times AUC_i}{\sum_{i=1}^{n} \#impression_i}$$

其中 n 是用户数，$\#impression_i$ 和 AUC_i 是第i个用户的impression数和对应AUC。

此外，引入了RelaImpr metric 去度量模型的相对提升：

$$RelaImpr = (\frac{AUC(measured\ model)}{AUC(base\ model) - 0.5} - 1) \times 100\%$$

4.4 Result from model comparison on Amazon Dataset and MovieLens Dataset

Table 3: Model Coparison on Amazon Dataset and MovieLens Dataset. All the lines calculate RelaImpr by comparing with BaseModel on each dataset respectively.

Model	MovieLens.		Amazon(Electro).	
	AUC	RelaImpr	AUC	RelaImpr
LR	0.7263	-1.61%	0.7742	-24.34%
BaseModel	0.7300	0.00%	0.8624	0.00%
Wide&Deep	0.7304	0.17%	0.8637	0.36%
PNN	0.7321	0.91%	0.8679	1.52%
DeepFM	0.7324	1.04%	0.8683	1.63%
DIN	**0.7337**	**1.61%**	**0.8818**	**5.35%**
DIN with Dice[a]	**0.7348**	**2.09%**	**0.8871**	**6.82%**

[a] Other lines except LR use PReLU as activation function.

结论：在上述两个数据集中，DIN表现最好。

原因：DIN关注用户兴趣的

DIN pays attentions to the locally related user interests by soft-searching for parts of user behaviors that are relevant to candidate ad.

因为此机制，DIN可以自适应得到用户兴趣的不同表示，极大提升了模型的表达能力；此外，DIN引入了Dice进一步提升了模型表现。

4.5 Performance of regularization

一共实验验证对比了其他4种正则技术：

- Dropout: Randomly discard 50% of feature ids in each sample
- Filter: Filter visited goods_id by occurrence frequency in samples and leave only the most frequent ones. In our setting, top 20 million goods_ids are left.
- Regularization in DiFacto: Parameters associated with frequent features are less over-regularized
- MBA: 即文中提出的Mini-Batch Aware正则技术

实验结果如下：

Table 4: Best AUCs of BaseModel with different regularizations on Alibaba Dataset corresponding to Fig.4. All the other lines calculate RelaImpr by comparing with first line.

Regularization	AUC	RelaImpr
Without goods_ids feature and Reg.	0.5940	0.00%
With goods_ids feature without Reg.	0.5959	2.02%
With goods_ids feature and Dropout Reg.	0.5970	3.19%
With goods_ids feature and Filter Reg.	0.5983	4.57%
With goods_ids feature and Difacto Reg.	0.5954	1.49%
With goods_ids feature and MBA. Reg.	**0.6031**	**9.68%**

结论：With goods_ids features and MBA 表现最好； 此外，Filter虽然表现比dropout效果好，但可能会丢失数据信息。

4.6 Result from model comparison on Alibaba Dataset

Table 5: Model Comparison on Alibaba Dataset with full feature sets. All the lines calculate RelaImpr by comparing with BaseModel. DIN significantly outperforms all the other competitors. Besides, training DIN with our proposed mini-batch aware regularizer and Dice activation function brings further improvements.

Model	AUC	RelaImpr
LR	0.5738	- 23.92%
BaseModel[a,b]	0.5970	0.00%
Wide&Deep[a,b]	0.5977	0.72%
PNN[a,b]	0.5983	1.34%
DeepFM[a,b]	0.5993	2.37%
DIN Model[a,b]	**0.6029**	**6.08%**
DIN with MBA Reg.[a]	**0.6060**	**9.28%**
DIN with Dice [b]	**0.6044**	**7.63%**
DIN with MBA Reg. and Dice	**0.6083**	**11.65%**

[a] These lines are trained with PReLU as the activation function.
[b] These lines are trained with dropout regularization.

结论：

- Taken together, DIN with MBA regularization and Dice achieves total 11.65% RelaImpr and 0.0113 absolute AUC gain over Base-Model；
- DIN, MBA 以及Dice均能进一步提升模型表现，三者结合时，模型表现达到最好；

4.7 Result from online A/B testing

在线A/B测试表明，其中DIN训练的模型带来10.0% CTR 和 3.8% RPM(Revenue Per Mille) 提升，相比于BaseModel.

4.8 Visualization of DIN

下图展示了用户兴趣分布：颜色越暖表示用户兴趣越高，可以看到用户的兴趣分布有多个峰。

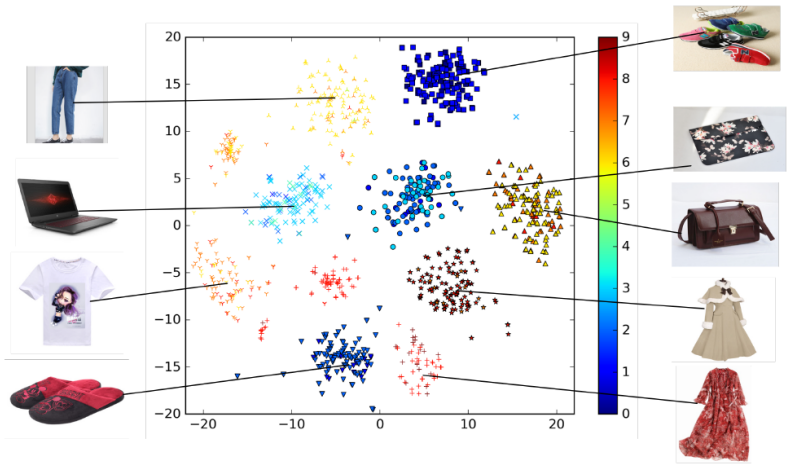


Figure 6: Visualization of embeddings of goods in DIN. Shape of points represents category of goods. Color of points corresponds to CTR prediction value.

利用候选的广告，反向激活历史兴趣。不同的历史兴趣爱好对于当前候选广告的权重不同，做到了local activation，如下图：

Figure 5: Illustration of adaptive activation in DIN. Behaviors with high relevance to candidate ad get high activation weight.

可以看到，对于候选的广告是一件衣服的时候，用户历史行为中跟衣服相关的权重较高，而非衣服的部分，权重较低。

5. 总结

1. 用户有多个兴趣爱好，访问了多个good_id，shop_id。为了降低纬度并使得商品店铺间的算术运算有意义，我们先对其进行Embedding嵌入。那么我们如何对用户多种多样的兴趣建模那？使用**Pooling对Embedding Vector求和或者求平均**。同时这也解决了不同用户输入长度不同的问题，得到了一个固定长度的向量。这个向量就是用户表示，是用户兴趣的代表。
2. 但是，直接求sum或average损失了很多信息。所以稍加改进，针对不同的behavior id赋予不同的权重，这个权重是由当前behavior id和候选广告共同决定的。这就是Attention机制，实现了Local Activation。
3. DIN使用*activation unit*来捕获local activation的特征，使用*weighted sum pooling*来捕获diversity结构。
4. 在模型学习优化上，DIN提出了*Dice激活函数*、*自适应正则*，显著的提升了模型性能与收敛速度。

参考文献

[1] Deep Interest Network for Click-Through Rate Prediction

[2] 盖坤的分享视频

[3] 探秘阿里之深度兴趣网络(DIN)浅析及实现

[4] 阿里Deep Interest Network理论